

What is claimed is:

- Sub A1
- 00200T-693-100200
- 1        1.        A method for holding up R-unit operands for a minimum number of cycles  
2 until all prior updates have completed by comparing addresses in at least one queue and  
3 interlocking valid R-unit register address matches, the method comprising:  
4                receiving a plurality of R-unit register addresses;  
5                storing said R-unit register addresses in a plurality of queues;  
6                accessing said queues;  
7                comparing said R-unit register addresses;  
8                determining matches between R-unit register addresses; and  
9                implementing one or more interlocks after said determining a valid match.
  - 1        2.        The method of claim 1 wherein said interlock causes a read instruction  
2 not to execute.
  - 1        3.        The method of claim 1 wherein said plurality of queues includes a  
2 write queue, pre-write queue and a read queue.
  - 1        4.        The method of claim 3 wherein a bypass sends an R-unit register addresses  
2 when said read queue is empty.
  - 1        5.        The method of claim 3 wherein said comparing includes comparing  
2 said R-unit register addresses sent to said read queue against said R-unit register  
3 addresses sent to said write queue.

POU920000162US1

002001"EEZ950

1           6.       The method in claim 3 wherein said determining includes matching a  
2 valid R-unit register addresses of said write queue and said read queue.

1           7.       The method in claim 3 wherein said determining includes matching  
2 said valid R-unit register addresses of said pre-write queue and said read queue.

1           8.       The method in claim 1 wherein said interlocks are implemented after  
2 said valid R-unit register address match is determined.

1           9.       The method in claim 1 wherein said interlocks prevent read  
2 instructions from being processed.

1           10.      The method in claim 1 wherein said write queue accumulates R-unit  
2 register addresses.

1           11.      The method in claim 1 wherein said updating occurs when an SRAM  
2 receives the accumulated results from said write queue.

00200T E9E22960

1           12.    A system for holding up R-unit operands for a minimum number of cycles  
2 until all prior updates have completed by comparing R-unit register addresses in at least  
3 one queue and interlocking valid R-unit register address matches, the system comprising:  
4                   a plurality of queues for storing R-unit register addresses;  
5                   a comparator for comparing said R-unit register addresses in said plurality  
6 of queues and determining matches between R-unit register addresses; and  
7                   a plurality of interlocks that are implemented after determining valid  
8 matches of said R-unit register addresses.

1           13.    The system of claim 12 wherein one of said interlocks causes a read  
2 instruction not to execute.

1           14.    The system of claim 12 wherein said plurality of queues includes a  
2 write queue, a pre-write queue, and a read queue.

1           15.    The system of claim 14 further comprising a bypass that sends an R-unit  
2 register addresses when said read queue is empty.

1           16.    The system of claim 14 wherein said comparator compares  
2 said R-unit register addresses sent to said read queue against said R-unit register  
3 addresses sent to said write queue.

1           17.    The system in claim 15 wherein said comparator determines said valid  
2 R-unit register address matches between said write queue and said read queue.

1 18. The system in claim 15 wherein said comparator determines said valid  
2 R-unit register address matches between said pre-write queue and said read queue.

1 19. The system in claim 13 wherein said interlocks are implemented after said  
2 valid R-unit register address match is determined.

1 20. The system in claim 13 wherein said interlocks prevent read instructions  
2 from being processed.

1 21. The system in claim 14 wherein said updating allows R-unit register  
2 addresses to accumulate in said write queue.

1 22. The system in claim 14 wherein said R-unit is updated when an SRAM  
2 receives the accumulated results from said write queue.

002001-0907-10000